In the Specification:

On page 1, after the Title of the Invention, please insert the following section:

RELATED APPLICATIONS:

The present application is a continuation of U.S. Application Serial No. 10/200,696 entitled INTERACTIVE TALKING DOLLS filed July 22, 2002, which is a continuation of U.S. Application Serial No. 09/883,762 entitled INTERACTIVE TALKING DOLLS filed June 18, 2001 and issued as U.S. Patent No. 6,497,604 on December 24, 2002, which is a continuation of U.S. Application Serial No. 09/685,527 entitled INTERACTIVE TALKING DOLLS filed October 10, 2000 and issued as U.S. Patent No. 6,309,275 on October 30, 2001, which is a continuation of U.S. Application Serial No. 08/831,635 entitled INTERACTIVE TALKING DOLLS filed April 9, 1997 and now abandoned.

Please amend the paragraph beginning on page 10, line 17 and ending on page 10, line 28 as follows:

In a preferred embodiment, at least ten input/output ports are provided so that the toy can perform at least five initiating actions and five responsive actions. However, it will be understood that because the number of input/output ports corresponds to the number of actions which may be performed, fewer or greater than ten ~~inlet/outlet~~ **input/output** ports may be provided depending on design choices. Thus, each microcontroller 110 preferably has six (6) pairs of input/output pins, five (5) of which are dedicated to random/sequential selection of an action (i.e., non-user determined selection of an action to be performed, the MCU 24 determining which action is to be performed

based on the setting of options setting 26). Of course, in the simplest form of the invention (in which a first toy performs an action and then activates a second toy to perform a responsive action, the action sequence ending upon completion of the responsive action) only a single input/output port is necessary.

Please amend the paragraph beginning on page 16, line 29 and ending on page 17, line 22 as follows:

The next step in the software program, or play subroutine, is for MCU A to generate a signal that causes the IR emitter to send a coded signal to the other doll (doll B) in step 218. This signal is coded to represent the appropriate responsive action that is to be performed by doll B. Doll A thus emits a signal that is received by doll B in step 220. The receipt of a signal wakes up doll B, whereas the completion of the performance of an action by doll A permits doll A to return to sleep. MCU B of doll B reads the coded signal emitted from doll A in step 222. Doll B then, in step 224, performs Action 2, shown separately in FIG. 5F. As with Action 1, Action 2 is shown separately because Action 2 represents a sub-subroutine that is performed at various points during the interactive play subroutine of FIGS. 5A-5D. Preferably, Action 2 represents the answering of the question asked by doll A. Typically, a single response is set for each question asked by the first-actuated doll. However, it is within the scope of the present invention to provide several answers to each of the questions asked, each answer either being randomly selected, sequentially selected, or user selected. The software randomly points at, or otherwise randomly selects, one of a plurality of codes (typically in a look up table, each code corresponding to a reaction-inducing action or a responsive action) set

by the program if the set option is in random. Alternatively, if the set option is in sequence, the software sequentially causes linear progression (such as by incrementation of a variable) through a set of actions that may be performed. Another option is to permit user selection with either a hard-wired or a remote control unit. Upon selection of the responsive action to be performed by the software program, Action 2 activates the output pin corresponding to the selected action code in step ~~400~~ **401** (FIG. 5F). As described above, the MCU is coupled to the voice chip via an output bus. Thus, the pin of the voice chip corresponding to the activated microcontroller pin is also activated, in step ~~402~~ **403**, to cause the speech pattern associated therewith to be enunciated by the voice chip.

Please amend the paragraph beginning on page 20, line 8 and ending on page 21, line 3 as follows:

The learn subroutine, implemented when MCU 24 is in learn mode so that a received infrared (or other wireless) signal from a wireless control device may be associated with a code for a desired action to be performed, will now be described with reference to Fig. 6. The number of buttons on the remote control device preferably corresponds to the number of actions the toy can perform, plus an additional button that corresponds to the hard-wired activation signal. Like the hard-wired activation signal, the additional button selects an action either randomly or in accordance with a preset sequence, depending on the doll's setting. Preferably six buttons are used for programming one doll and a different six buttons are used for programming the other doll. In step 400 of the learn subroutine shown in Fig. ~~5~~ **6**, the learn software subroutine is started. The user points a remote control first at one doll and then at the other doll and sequentially presses the number of remote

control buttons necessary to correlate with each action to be performed so that the dolls can be programmed to respond differently to the pressing of each of the buttons. Thus, the buttons used for one doll are different from the buttons used for the other doll. Each time a user presses a button of the remote control unit, the MCU of the doll being programmed reads the signal in step 402. Before continuing, the MCU must determine, in decision step 404, whether the received signal is valid (recognizable by the MCU). If not, the MCU learn subroutine returns to step 404 to read another signal. If the signal, however, is valid, then the subroutine continues with step 406, in which the read signal is saved in a predefined address (associated with one of the possible actions) in the program for later use. After saving the signal, decision block 408 determines whether all coding buttons have been programmed. If not, the subroutine returns to step 402 to read another signal from the remote control. Once all of the buttons have been programmed, there are no more addresses to be assigned with a coded signal and the subroutine continues with step 410, in which the MCU rests until activated by one of the above-described actuation signals. It will be appreciated that fewer or greater than six buttons may be programmed, depending on the number of actions that may be performed.